EXHIBIT 8

DOCKET NO: 0100157-00244

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT: 6,415,280
INVENTOR: DAVID A. FARBER
AND RONALD D. LACHMAN
FILED: APR. 1, 1999 ISSUED: JUL. 2, 2002
TITLE: IDENTIFYING AND
REQUESTING DATA IN A
NETWORK USING IDENTIFIERS
WHICH ARE BASED ON THE
CONTENT OF THE DATA

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES REVIEW* OF U.S. PATENT NO. 6,415,280 UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES

I. MANDATORY NOTICES

A. Real Parties-in-Interest

EMC Corporation and VMware, Inc. ("Petitioner") are the real parties-in-interest.

B. Related Matters

The '280 patent is one of an extensive patent family of continuation and divisional applications. Exhibit 1008 shows the patent family, with patents in red and blue including the '280 patent being asserted in the litigation *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.* (No. 6:11-cv-00660-LED) (E.D. Tex.), served on December 16, 2012.

Petitioner is also seeking Inter Partes Review of related U.S. Patents Nos. 5,978,791, 7,945,539, 7,945,544, 7,949,662, and 8,001,096, and requests that they be assigned to the same Board for administrative efficiency. Moreover, there are several continuing applications related to this family that remain pending (shown on Exhibit 1008 in green). Because they share a common disclosure with the '280 patent, these applications may be used as a basis to present patentably indistinct claims that may issue prior to the determination of the PTAB in this or related Inter Partes Reviews. The issuance of indistinct claims is at least inconsistent with Rule 37 C.F.R. 42.73(d)(ii) and would be an "end-around" the reasonable number of substitute claims that are permitted in an IPR proceeding. Petitioner respectfully requests that the PTAB suspend from further prosecution, *sua sponte*, the

applications in this related family, including the applications shown on Ex. 1008 in green and any further applications that may be filed that depend from this family of patents. If the PTAB determines that that suspension should be requested by written motion, permission to file such a motion is requested at this time.

C. Counsel

Lead Counsel: Peter M. Dichiara (Registration No. 38,005)

Backup Counsel: David L. Cavanaugh (Registration No. 36,476)

Petitioners will request authorization to file a motion for Cynthia Vreeland to appear *pro hac vice*. Ms. Vreeland has more than 20 years litigation experience, and has worked with Petitioner EMC on IP litigation matters for more than 10 years. As such, Ms. Vreeland has experience and established familiarity with the technology at issue in the case. Petitioners intend to file a motion seeking admission of Ms. Vreeland to appear *pro hac vice* when authorized to do so.

D. Service Information

Email: Peter Dichiara, Peter.Dichiara@wilmerhale.com

Post and Hand Delivery: WilmerHale, 60 State St., Boston MA 02109

Telephone: 617-526-6466 Facsimile: 617-526-5000

E. CERTIFICATION OF GROUNDS FOR STANDING

Petitioner certifies pursuant to Rule 42.104(a) that the patent for which review is sought is available for *inter partes* review and that Petitioner is not

barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

II. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

A. Prior Art Patents and Printed Publications

Pursuant to Rules 42.22(a)(1) and 42.104 (b)(1)-(2), Petitioner challenges claims 36 and 38 of U.S. Patent No. 6,415,280 ("the '280 patent"; Ex. 1001) as anticipated by or unpatentable in view of the following patents and printed publications:

1. S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," University of Tennessee Technical Report CS-95-278 (Feb. 1995) ("Browne ", Ex. 1002).[1]

---

[1]The Browne February 1995 publication qualifies as prior art under 35 USC 102(a), and is used in this petition because it includes illustrations facilitating explanation of the invalidity of the challenged claims. Petitioner also has attached as exhibits and included in its claim charts two earlier versions of this publication – S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," http://www.netlib.org/utk/papers/lifn/main.html (Nov. 11, 1994) (Exhibit 1006); and K. Moore et al., "An Architecture for Bulk File Distribution," Network Working Group Internet Draft (July 27, 1994) (Exhibit 1007). As Dr. Clark confirms in his Decl., the references are substantially the same with respect to the disclosure relevant to the challenged claims of the '280 patent. If the Patent

2. Woodhill et al., U.S. Patent No. 5,649,196, entitled "System and Method For Distributed Storage Management on Networked Computer Systems Using Binary Object Identifiers," filed Nov. 9, 1995 as a continuation of application 85,596, filed July 1, 1993 ("Woodhill", Ex. 1005).
3. Legent Software, Inc., "ESM: Product Information," Legent Corporation (April 1994) ("ESM Manual", Ex. 1026).[2]
4. Satyanarayanan, "Scalable, Secure, and Highly Available Distributed File Access," *IEEE Computer*, vol. 23, no. 5 (May 1990), pp. 9–21 ("Satyanarayanan," Ex. 1029)
5. Albert Langer, "Re: dl/describe (File descriptions),")," article

---

Owner alleges an earlier priority date of the challenged claims, Petitioner may rely on the earlier publications for invalidity, alone or in combination with the other references cited in this petition.

[2] The ESM manual was published by Legent Corporation in April 1994. This manual was accessible to the public and made available by Legent sales personnel to anyone that expressed interest in the Enterprise Storage Manager product. The purpose of the manual was to promote the sales of Enterprise Storage Manager by describing its operation and usefulness, and no purchase was required to receive this manual. As such, the manual was freely accessible to anyone in the relevant community of distributed file system and network administrators.

<1991Aug7.225159.786@newshost.anu.edu.au> in Usenet newsgroups "alt.sources.d" and "comp.archives.admin" (August 7, 1991) ("Langer", Ex. 1003)

6. Kantor, "The Frederick W. Kantor Contents-Signature System Version 1.22," FWKCS122.REF (August 10, 1993) ("Kantor", Ex. 1004).[3]

B. There is a Reasonable Likelihood that at least One Claim of the '280 Patent is Unpatentable Under 35 U.S.C. §§ 102, 103

Section VI below explains how the above-cited patents and printed publications create a reasonable likelihood that Petitioner will prevail with at least one of the challenged claims. *See* 35 U.S.C. § 314(a). Indeed, that section supported by the attached claim charts of Exhibits 1032 through 1036 and the Declaration of Dr. Douglas Clark, a Professor of Computer Science at Princeton

[3] Kantor's FWKCS user manual has been publicly and freely available continuously since August 1993. Kantor distributed the user manual with the FWKCS program as shareware and posted it online to electronic Bulletin Board Systems including "The Invention Factory" and "Channel 1" for an extended period of time, where it could be downloaded by anyone. As such the document was accessible to others in the relevant community of BBS users and system operators. (*See* Kantor at 3, 5; *see also* 158-59; Ex. 1004.)

University ("Clark Decl."; Ex. 1009), demonstrate that all of the challenged claims are anticipated by, or unpatentable in view of, each of these references.

C. Relief Requested

Petitioner requests cancellation of claims 36 and 38, the challenged claims, as unpatentable under 35 U.S.C. §§ 102 and 103.

III. Claim Construction

The claim terms should be given their "broadest reasonable construction in light of the specification." 37 C.F.R. § 42.100(b).

The claim terms can be understood by their ordinary and plain meanings except where construed in the specification. The specification includes the following constructions relevant to the challenged claims:

Term	Construction
"data" and "data item"	"as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits. ('280 patent, col. 1, ll. 53-59, s*ee also* col. 1, l. 64 – col. 2, l. 2 ("data items (the data items being files, directories, records in the database, objects in an object-oriented programming, locations in memory or on a physical device or the like)"); Ex. 1001.)

Term	Construction
"file system"	"a collection of directories. A directory is a collection of named files – both data files and other directory files" ('280 patent, col. 5, ll. 45-47; Ex. 1001.)
"file"	"a named data item which is either a data file (which may be simple or compound) or a directory file. A simple file consists of a data segment. A compound file consists of a sequence of data segments. A data segment is a fixed sequence of bytes" ('280 patent, col. 5, ll. 47-55; Ex. 1001.)
"location"	"with respect to a data processing system, refers to any of a particular processor in the system, a memory of a particular processor, a storage device, a removable storage medium (such as a floppy disk or compact disk), or any other physical location in the system" ('280 patent, col. 5, l. 66- col. 6, l. 4; Ex. 1001.)
"local"	"with respect to a particular processor refers to the memory and storage devices of that particular processor" ('280 patent, col. 6, ll. 4-5; Ex. 1001.)
"True Name, data identity, and data identifier"	"refer to the substantially unique data identifier for a particular item" ('280 patent, col. 6, ll. 7-11, s*ee also* col. 14, l. 52-col. l. 24 (describing mechanism for calculating True Name using MD hash function); Ex. 1001.)

IV. OVERVIEW OF THE '280 PATENT

A. Brief Description

The '280 patent is directed to data storage systems that use "substantially unique data identifiers" – based on all the data in a data item and only the data in the data item – to identify and access data items. (*See, e.g.*, '280 patent, Title, Abstract, and col. 1, ll. 12-17; Ex. 1001.) The patent uses these identifiers to perform basic file management functions, such as storing and retrieving replicated copies of computer files or other data items, and eliminating unwanted and unnecessary duplicate copies of these items—admittedly old problems. (*See*, e.g., '280 patent, Background of the Invention, col. 2, l. 45-56; Ex. 1001.)

According to the patent, prior art systems identified data items based on their location or address within the data processing system. ('280 patent, col. 1, ll. 22-27; Ex. 1001.) For example, files were often identified by their context or "pathname," that is, information specifying a path through the computer directories to the particular file (e.g., C:\My Documents\Law School\1L\TortsOutline.txt). ('280 patent, col. 1, ll. 34-41; Ex. 1001.) The patent contends that all prior art systems operated in this manner: "In ***all*** of the prior data processing systems the names or identifiers provided to identify data items. . . are ***always*** defined relative to a specific context," and "there is ***no*** direct relationship between the data names and the data item." ('280 patent, col. 1, l. 64 – col. 2, l. 2, col. 2, ll. 11-12, emphasis added; Ex. 1001.)

According to the patent, this prior art practice of identifying a data item by its context or pathname resulted in certain shortcomings. For example, with pathname identification, the same data name may refer to different data items, or conversely, two different data names may refer to the same data item. ('280 patent, col. 2, ll. 11-15; Ex. 1001.) Moreover, because there is no correlation between the contents of a data item and its pathname, there is no *a priori* way to confirm that the data item is in fact the one named by the pathname. ('280 patent, col. 2, ll. 16-19; Ex. 1001.) Furthermore, context or pathname identification may more easily result in the creation of unwanted duplicate data items, e.g., multiple copies of a file on a file server.[4] ('280 patent, col. 2, ll. 46-57; Ex. 1001.)

The '280 patent purports to address these shortcomings. ('280 patent, col. 3, ll. 5-19; Ex. 1001.) It suggests that "it is therefore desirable to have a mechanism . . . to determine a common and substantially unique identifier for a data item, using only the data in the data item and not relying on any sort of context." ('280 patent, col. 3., ll. 5-10; Ex. 1001.) Moreover, "[i]t is further desirable to have a mechanism for reducing multiple copies of data items… and to have a mechanism

[4] For example, Alice and Bob both download the same copy of the James Bond movie *Goldfinger*. Alice saves her copy at "C:\Movies\Bond\Goldfinger.mov", and Bob saves his copy at "C:\Videos\007\Bond-Goldfinger.mov".

which enables the identification of identical data items so as to reduce multiple copies." ('280 patent, col. 3, ll. 11-16; Ex. 1001.)

To do so, the '280 patent provides data identifiers that "depend[] on all of the data in the data item and only on the data in the data item." ('280 patent, col. 1, ll. 11-16, col. 3, ll. 28-31; Ex. 1001.) Preferred embodiments use either of the well-known MD5 or SHA message digest functions[5] to calculate a substantially unique identifier from the contents of the data item. ('280 patent, col. 12, l. 38- col. 14, l. 24; Ex. 1001.) The system first computes the 16-byte (128-bit) message digest of the data item and then appends the size of the data item to produce a 160-bit identifier. ('280 patent, Fig. 10A and col. 13, ll. 52-63; Ex. 1001.) The patent calls these context- or location-independent, content-based identifiers a "True Name" – a phrase admittedly "coined by the inventors." (U.S. Patent No. 6,415,280 Prosecution History, Response (Aug. 22, 2001), at 22; Ex. 1019.)

---

[5] A message digest function is a transformation of a piece of data into a much shorter form. (*See, e.g.*, D. Banisar et al., The Third CSPR Cryptography and Privacy Conference at 509 (1993); Ex. 1010 (describing a message digest function as "a 128-bit cryptographically strong one-way hash function of the message" that is "somewhat analogous to a 'checksum' or CRC error checking code, in that it compactly 'represents' the message.").) The '791 patent admits that message digest functions were known. ('280 patent, col. 12, l. 66 - col. 13, l. 3; Ex. 1001.)

With these identifiers, the patent asserts, "data items can be accessed by reference to their identities (True Names) independent of their present location." ('280 patent, col. 34, ll. 20-22; *see also* col. 34, ll. 41-43; Ex. 1001.) The actual data item corresponding to these location-independent identifiers may reside anywhere, e.g., locally, remotely, offline. (*Id.* at col. 34, ll. 22-30.) "Thus the identity of a data item is independent of its name, origin, location, address, or other information not derivable directly from the data, and depends only on the data itself." (*Id.* at col. 3, ll. 31-33.)

In the preferred embodiments, the substantially unique identifiers are used to "augment" standard file management functions of an existing operating system. (*Id.* at col. 6, ll. 12-19.) For example, a local directory extensions (LDE) table[6] is indexed by a pathname or contextual name of a file and also includes True Names for most files. (*Id.* at col. 8, ll. 24-31.) A True File registry (TFR) lists True Names, and stores "location, dependency, and migration information about True Files." (*Id.* at col. 8, ll. 32-34, 37-39.) True Files are identified in the True File registry by their True Names, and can be looked up in the registry by their True

---

[6] According to the patent, a LDE table is a data structure which provides information about files and directories in the system and includes information in addition to that provided by the native file system. ('280 patent, col. 8, ll. 24-31; Ex. 1001.)

Names. (*Id.* at col. 8, ll. 35-37; col. 23, ll. 50-51.) This look-up provides, for each True Name, a list of the locations, such as file servers, where the corresponding file is stored. (*Id.* at col. 34, ll. 28-30; *see also* col. 15, ll. 62-64.) When a data item is to be "assimilated" into the system, its True Name can be compared to the True File Registry to see if the data item already exists in the system. (*Id.* at col. 14, ll. 25-32.) The True Name also can be used to identify a file by contents, to confirm that a file matches its original contents, or to compare two files." (*Id.*at col. 15, ll. 9-11.)

The system also includes a "Mirror True File" background mechanism "to mirror (make copies) of the True File available elsewhere in the system." ('280 patent, col. 35, ll. 15–18; Ex. 1001.) "In operation data items can be accessed by reference to their identities (True Names) independent of their present location. The actual data item or True File corresponding to a given data identifier or True Name may reside anywhere in the system (that is, locally, remotely, offline, etc)." (*Id.* at col. 34, ll. 21–24.) If a data item is not present locally, the True File registry may be used to determine the location(s) of copies of the True File corresponding to a given True Name. (*Id.* at col. 34, ll. 27-31.) Another mechanism, the "Realize True File from Location" primitive mechanism, "tries to make a local copy of a True File, given its True Name and the name of a source location (processor or media) that may contain the True File." (*Id.* at col. 34, ll. 31-35; *see also* col. 16,

ll. 17-19.) If the source location is a remote processor, the Realize True File from Location mechanism sends a message to that remote processor and waits for a response. (*Id.* at col. 15, l. 66 -col. 16, l. 3.) If it receives a positive response, it enters the returned True File into the True File Registry. (*Id.* at col. 16, ll. 5-7.)

B. The Prosecution History of the '280 Patent

The '280 patent is based on an application that was originally filed on April 11, 1995, and is a division of the application that resulted in U.S. Pat. No. 5,978,791. The claims were preliminarily amended to include new claims 54, 87 and 89 (among others), which are reproduced below:

--54. (New) In a system in which a set of data items are distributed across a network of servers, at least some of the data items being cached versions of data items from a source server, a content delivery method comprising:

determining a data identifier for a particular data item, the data identifier being determined using a given function of the data comprising the particular data item; and

responsive to a request for the particular data item, the request including at least the data identifier of the particular data item, providing the particular data item from a given one of the servers of the network of servers.

****

87. (New) A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:

storing the data item is on a first server in the network and storing copies of the data item on a set of servers in the network distinct from the first server; and

responsive to a client request for the data item, the request including a hash of the data item, causing the data item to be provided to the client.

****

89. (New) A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:

storing the data item is on a first server and storing copies of the data item on a set of servers distinct from the first server; and

responsive to a client request for the data item, the request including a value determined as a given function of the data in the data item, providing the data item to the client.

(Prelim. Amend. Of Jan. 11, 2001, at 2 and 11, Ex. 1024.)

All of the pending claims were rejected for failing to satisfy 35 U.S.C. §112, ¶1 and as unpatentable over Nelson (U.S. Pat. No. 5,452,447) in view of Hamilton (U.S. Pat. No. 5,640,564) (Office Action of June 5, 2001 at 4 and 6, Ex. 1025). In connection with the latter, the office action primarily focused on claim

54, and relied on Nelson for the basic components of the recited system and on Hamilton for the recited data identifier:

> As to claims 54-56, Nelson substantially discloses the invention including a data processing system for caching a file server to thereby allow client users to request and retrieve files in a distributed computer system (abstract, lines 1-8 et seq). In particular, Nelson discloses a plurality of network servers (fig. 3, items 56, 58, 60, 68) including at least some cached data items from a source server (see fig. 1, items 28, 30; abstract, lines 8-15, et seq). Nelson further discloses the use of a hash function on a data file to thereby quickly retrieve the data file from the cache upon user's request (col.17, lines 18-41 et seq).
>
> Nelson does not particularly detail the use of the hashing function on the data file to create an identifier, which can be utilized to retrieve the data file upon user's request. However, Hamilton discloses an analogous system wherein a hashing function is applied to a data item to thereby create an identifier, which a user can utilize to request and retrieve a corresponding data item (col. 6, lines 28-39 et seq). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references because Hamilton's teaching would allow the users of Nelson's system to expeditiously and dynamically retrieve a file as it is updated.

(*Id*. at 7; Ex. 1025.) With regard to claims 87 and 89, the office action stated, without separate discussion, that the rejection was based on similar grounds:

16. The limitations of claims 62-106 have already been discussed in the in the rejection of claims 54-61 above . They are therefore rejected on similar grounds.

(*Id*. at 8 Ex. 1025.) In reply to the §112 rejections, the applicants argued:

> An important aspect of this invention is the so-called *True Name*—a term coined by the inventors of this invention. The "terms 'True Name', 'data identity' and 'data identifier' refer to the substantially unique data identifier for a particular data item.
>
> ****
>
> Thus, the application teaches accessing data items using their True Names (e.g., hashes of their contents). And it further teaches accessing data items (using their True Names) from any location and independent of the location of the data items. Further, using a data item's True Name, the data item may be obtained from one or more locations, e.g., as specified in a True File registry table. As discussed above, the True File Registry table may contain "source ID(s) of . . . sources from which this file or data item may be retrieved." *Specification*, pg. 17, liens 10-12.
>
> The application describes, for that embodiment, using the mechanism *Realize True File from Location* to obtain the requested data item. The *Realize True File from Location* "mechanism is used to try to make a local copy of a True File, given its True Name and the name of a source location (processor or media) that may contain the True File." *Specification*, pg. 29, lines 13-16. Note that this mechanism is described in detail at pg. 29, lines 12 to pg. 30, line 5 and with reference to FIG. 15.

(Response, Aug. 22, 2001 at 22-23; Ex. 1019.)

In addition, the applicants submitted claims charts, purportedly identifying portions of the specification that they alleged taught and supported claims 87 and 89, among others. (*Id.* at 24. Ex. 1019.) Those charts are reproduced below:

Claim 87	Support in Specification
A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:	See generally Fig. 1., and description at pg. 8, lines 8 to pg. 9, line 3. ", the processors may be in one of various relationships. For example, two processors 102 may be in a client/server, client/client, or a server/server relationship. . . ." pg. 8, lines 21-24.
storing the data item is on a first server in the network and storing copies of the data item on a set of servers in the network distinct from the first server; and	See generally the mechanism *Mirror True File* at pg. 50 et seq. ". . . use to ensure that files are available in alternate locations." Pg. 50, lines 16-18.
responsive to a client request for the data item, the request including a hash of the data item, causing the data item to be provided to the client.	pg. 66, line 16 to pg. 67, line ___. *Realize True File from Location*, pg. 29, lines 12 to pg. 30, line 5 and FIG. 15. *Request True File*, pg. 46, lines 7-24. "functions . . . include MD4, MD5, and SHA" pg. 23, lines 11-12. These are known hash functions.

****

Claim 89	Support in Specification
A method of delivering a data item in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:	See generally Fig. 1., and description at pg. 8, lines 8 to pg. 9, line 3. ", the processors may be in one of various relationships. For example, two processors 102 may be in a client/server, client/client, or a server/server relationship. . . ." pg. 8, lines 21-24.
storing the data item is on a first server and storing copies of the data item on a set of servers distinct from the first server; and	See generally the mechanism *Mirror True File* at pg. 50 et seq. ". . . use to ensure that files are available in alternate locations." Pg. 50, lines 16-18.
responsive to a client request for the data item, the request including a value determined as a given function of the data in the data item, providing the data item to the client.	pg. 66, line 16 to pg. 67, line ___. *Realize True File from Location*, pg. 29, lines 12 to pg. 30, line 5 and FIG. 15. *Request True File*, pg. 46, lines 7-24.

(*Id.* at 27 and 38-39; Ex. 1019.)

In connection with the prior art rejections, the applicants amended the claims to specifically recite "data files" as opposed to "data items," and to recite that the

request included a hash of the "contents of" the data item, as opposed to merely a hash of the data item. (*Id.* at 47-58) The applicants' arguments focused on the purported novelty of the content- based identifier:

> The claims have been amended to clarify that the identifier determined for a data identifier is context sensitive, i.e., is based on the content of the data or file

(*Id.* at 24; Ex. 1019.) Nowhere did the applicants dispute that Nelson taught the basic system of servers storing copies of files or causing the file to be provided in response to a request.

The claims were subsequently allowed, and claims 87 and 89 were renumbered to challenged claims 36 and 38, respectively.

V. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. There is Nothing New About Using Content-Based Identifiers to Request and Obtain a Data File from a Network

The '280 claims focus on the concept of using content-based identifiers to store, request and obtain copies of data files from a set of servers. Claim 36, for example, simply requires storing copies of a data file on multiple servers in a network, and causing a copy of the file to be provided to a client in response to a request including a hash of the contents of the file:

> 36. A method of delivering a data file in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:

> storing the data file is [sic] on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server; and
>
> responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client.

('280 patent, col. 43, ll. 54-64; Ex. 1001.)[7]

The applicants indicated in their patent application that they were entitled to these broad claims because "[i]n ***all*** of the prior data processing systems the names or identifiers provided to identify data items . . . are ***always*** defined relative to a specific context," and "there is ***no direct relationship*** between the data names and the data item." ('280 patent, col. 1, l. 64–col. 2. l. 2, col. 2, ll. 11-12, emphasis added; Ex. 1001.)

These representations were simply wrong. Prior data processing system ***did use*** identifiers based on the content of a data item itself, and not its context or pathname. In fact, these techniques were old and widely used. This is not surprising. The concept of using a mathematical function to create a "fingerprint" or "signature" for a data item based on the content of the data item predates the

[7] Claim 38 is similar but replaces the reference to a "hash" with the amorphous phrase "value determined as a given function."

'791 patent by decades. For example, IBM developed one of the first hash tables in the 1950s (*see*, *e.g.*, D. Knott, Hashing functions, The Computer Journal 18 (1975), vol. 3, at 274 (discussing "history of hashing"); Ex. 1011), and Professor Ron Rivest of MIT introduced the MD5 algorithm referenced in the '791 patent in the early 1990s. (*See, e.g.*, R. Rivest, "The MD5 Message-Digest Algorithm," Internet RFC 1321 (Apr. 1992); Ex. 1012.) These hashing functions take as input the data contained in a file or other data item, and produce a much smaller-sized output value, commonly called a "hash," "hash value," "message digest" ("MD"), or "checksum." (*See, e.g.*, McGraw-Hill Dictionary of Scientific and Technical Terms, (4th ed., 1989), at 860; Ex. 1013; *see also* B. Kaliski, "A Survey of Encryption Standards," IEEE Micro (Dec. 1993), pp. 74–81, at 77; Ex. 1014.) For example, a file that is a million bytes (or even much larger) in size can be used as input to produce a hash value that is a mere 16 bytes in length. Because of the mathematical properties of the function, the odds that two different files will produce the identical 16 byte hash are extremely small: for example, with a 16 byte hash output, the odds that two randomly picked inputs have the same hash are 2^{-64}, or approximately one in sixteen billion billions. (B. Kaliski at 77; Ex. 1014.) Consequently, hashes are known as "signatures" or "fingerprints" because they identify data items with high reliability, just like signatures or fingerprints are used to identify people with a high degree of certainty. (*See* D.R. McGregor and J.A.

Mariani, 'Fingerprinting' – A Technique for File Identification and Maintenance, Software Practice & Experience 1165 (1982), vol. 12, no. 12, at 1165 ("fingerprinting" technique "produce[s] a quasi-unique identifier for a file, derived from that file's contents. . .[t]he idea is to provide an identifying deature for every file, which is intrinsically distinctive, and analogous (hopefully) to a human's fingerprint."); Ex. 1017.)

Although applicants suggested in their application that they were the first to utilize these hashing functions to identify data items for file management applications, others working in the field used them for the same purposes more than a decade before the '280 patent. For example, file "fingerprinting" has long been known as a technique to see if two files are identical. (*See* Rabin, Fingerprinting by Random Polynomials, Center for Research in Computing Technology, Harvard University, Report TR-15-81 at 1 and 9, 1981; Ex. 1015; *see also* Manber, at 3 (commenting on work of Rabin); Ex. 1016). Likewise, the use of "fingerprints" both to identify files and to check for duplicates also has long been known. (S*ee, e.g.*, D.R. McGregor and J.A. Mariani, at 1165; Ex. 1017.)[8]

[8] This reference was cited and central to the analysis and rejection of EP counterpart application EP0826181A1 with claims having a central feature of content-based identifiers. (Annex to the communication, May 8, 2009; Ex. 1020.) Applicant amended the claims to emphasize a "licensing" limitation not found in

Many printed publications and patents disclose and use data identifiers exactly like those described and claimed in the '280 patent for exactly the same purposes. These publications disclose identifiers that are location- and context-independent, that are determined based on hash of the contents of a data file, and that are formed using the same algorithms mentioned in the '791 patent.

Browne: For example, researchers at the University of Tennessee and Bell Laboratories disclosed a system that created "location-independent file names" (or "LIFNs") to identify files on the Internet. (Browne at 3; Ex. 1002.) LIFNs – like the identifiers in the '280 patent – uniquely identified files by their contents, not their locations. (Browne at 3; Ex. 1002; *compare* '280 patent, col. 34, ll. 21-23 (True Names used to identify files "independent of their present location"); Ex. 1001.) LIFN signatures were computed as "the ascii form of the MD5 signature of the file" – the same function identified in the '280 patent. (Browne at 6; Ex. 1002; *compare* '280 patent, col. 12, l. 66- col. 13, l. 1, (MD5 or SHA used to calculate True Name); Ex. 1001.) Browne specifically addressed mirrored file repositories,

---

the challenged claim (Reply to communication from the Examining Division, Nov. 18, 2009 at 4; Ex. 1021), but this too was found unpersuasive and the rejection was maintained by the EPO. (Annex to the communication, March 14, 2012 at 4; Ex. 1022.) Following this rejection, Applicants withdrew the application from consideration. (Closing of Application, June 14, 2012; Ex. 1023.)

where copies of a file were replicated to multiple secondary servers in a network: "copies of popular software packages may be mirrored by a number of sites to increase availability (e.g., if one site is unreachable, the software may be retrieved from a different site) and to prevent bottlenecks." (Browne at 2; Ex. 1002.) A client could use a LIFN to obtain a list of file servers, or "mirror sites," that could provide a copy of the file corresponding to the LIFN. (Browne at 4–5; Ex. 1002.) The client could then select any of the servers from the list, and use the LIFN to contact the server and to request and obtain a copy of the corresponding file. (Browne at 4, 6; Ex. 1002.)

Woodhill: Woodhill provides another example of the use of context- and location-independent identifiers for the same purposes as the '280 patent. Woodhill created a distributed storage system that used "Binary Object Identifiers" to store, request, and obtain copies of files comprised of binary objects, among other functions. (Woodhill; Ex. 1005) Woodhill stored multiple copies of binary objects using a "Concurrent Onsite/Offsite Backup" process that stored copies of each binary object in at least two locations in addition to the local computer on which it normally resided: (1) one copy was stored "somewhere on the local area network 16 other than on the local computer 20 on which it normally resides" and (2) another copy was stored on a remote backup file server. (Woodhill at col. 9, ll. 29-44 ; Ex. 1005.)

Like the '280 patent, a client request for a binary object, identified by a Binary Object Identifier, included a hash of the contents of the binary object. As Woodhill explains, a "Binary Object Identifier 74 [of Fig. 3] . . . is a unique identifier for each binary object to be backed up." (Woodhill at col. 4, lines 45-47; Ex. 1005). "Each of the fields of the Binary Object Identifier 74 . . . is calculated from the contents of each binary object." (Woodhill at col. 8, lines 1-4; Ex. 1005). Among these fields is a "Binary Object Hash field 70" that "is calculated against the contents of the binary object." (Woodhill at col. 8, lines 21-23; Ex. 1005.) A binary object could be requested by its Binary Object Identification Record, including the Binary Object Identifier with Binary Object Hash field. (Woodhill at Figure 3 and col. 10, ll. 27-34; Ex 1005.) For example, files on remote backup servers were periodically audited by "initiat[ing] a restore of a randomly selected binary object identified by a Binary Object Identification Record 58." (Woodhill at col. 18, ll. 11-19; Ex. 1005.) In response to such a request, the binary objects would be provided to the client local computer from which the request originated. (Woodhill at col. 18, ll. 19-23; Ex. 1005.)

ESM Manual: The ESM Manual provides yet another example of a system that used context- and location-independent identifiers for the same purposes as the '280 patent. The ESM manual describes the Enterprise Storage Manager (ESM) product. ESM provided a network backup solution that "protects data by

completely automating the backup process, by saving the data in two locations, on the LAN and at the corporate data center," in addition to the copy saved on the local server (i.e., three total copies). (ESM Manual at 2-1; Ex. 1026.) ESM, like Woodhill, subdivided files into binary objects or "BOBs." (ESM Manual at 3-4; Ex. 1026.) Each BOB was identified by its BOB identifier or "BOBID," which was "calculated from the contents of the BOB itself," and included a 32-bit hash of the contents of the BOB. (ESM Manual at 3-4; Ex. 1026.) Like the '280 patent, a client local computer could request a BOB (e.g., during restore operations) using its BOBID including the hash of the BOB. (ESM Manual at 3-9; Ex. 1026.) In response to such a request, the file would be obtained and provided to the client. (ESM Manual at 5-3; Ex. 1026.)

Satyanarayanan in view of Langer or Kantor: The fact that multiple prior art references disclose the use of identifiers including hashes for the same purposes as the '280 patent – storing, requesting and obtaining copies of mirrored or replicated data files – is hardly surprising. This is an obvious use of context- or location-independent identifiers. The '280 patent itself admits that file mirroring technology is old. (**'**280 patent at col. 2, l. 58 to col. 3, l. 2; Ex. 1001.) During prosecution, the Patent Office found that Nelson taught this structure, and the applicants did not dispute it. Nor could they – replicated file systems predated the '280 patent by decades. (*See, e.g.,* "A Principle for Resilient Sharing of

Distributed Resources," published in 1976, at 1 and 19; Ex. 1030; and "NonStop System," published in 1978, at 1; Ex. 1031.) Multiple prior art references, including Langer and Kantor, in addition to Browne, Woodhill, and the ESM manual, confirm that identifying files with hashes of their contents was likewise old by the time of the '280 patent. Langer and Kantor, like Browne, Woodhill, and the ESM manual, used these techniques for precisely the same reason as the '280 patent, *i.e.*, content-based, location-independent substantially unique identification of a data file. A person of ordinary skill in the art would have found it obvious to apply the teachings of Langer or Kantor to Satyanarayanan to enable client requests for files that use content-based identifiers including a hash of the contents of a file. Langer and Kantor provide express motivation to make this combination, by disclosing that the use of a unique, location-independent identifier allows a client to automatically access a file from the nearest location, or to access a file with an inherent integrity check. As further discussed *infra,* such a combination of Satyanarayanan with Langer or Kantor would have been the application of Kantor's or Langer's known techniques to the known device of Satyanarayanan, ready for improvement, to yield the predictable result of improving file access using unique identifiers. (*See* Langer at 3–4; Ex. 1003; Kantor at 6; Ex. 1004.)

These prior art references provide just a handful of many examples of the use of content-based identifiers to store, request and obtain data files. Indeed, the

application of hash-based identifiers to these functions was so obvious that at least one commentator not only described the applications as "easy" but also posted these ideas publicly "to impede anyone who might independently have had the idea from patenting it." (Williams, "An algorithm for matching text (possibly original)", posted to the "comp.compression" newsgroup on January 27, 1992; Ex. 1027; *see also* R. Williams, "An Introduction to Digest Algorithms," Rocksoft (Nov. 1994), at 13 ("digest algorithms can be used . . . to generate unique fixed-length identifiers for arbitrary blocks of data in situations where the identifier of identical blocks must be the same" and that "a network of computers implementing a distributed database of documents could collect documents . . . and then synchronise at night by exchanging and comparing the digests of the new documents"); Ex. 1028).

In short, other than perhaps coining a new phrase – i.e., True Name – for a very old concept, there is absolutely nothing new disclosed or claimed in the '280 patent concerning the use of location-independent, content-based data identifiers.

Pursuant to Rule 42.104(b)(4)-(5) and Practice Guide Fed. Register Vol. 77, No. 157, page 48764, Petitioners have submitted claim charts in connection with this Petition (attached as Exhibits 1032 through 1036), from the pending litigation between the Petitioner and PersonalWeb Technologies LLC which claims to be the proper owner of the '280 patent. Those charts confirm that the challenged claims

are indeed anticipated and/or unpatentable in view of the cited references. Moreover, Petitioners also submit the Declaration of Dr. Douglas Clark (Ex. 1009), a Professor of Computer Science at Princeton University, who has considerable experience in industry and academia. In his declaration, Dr. Clark confirms that those charts identify representative subject matter in each reference that teaches each and every limitation of the challenged claims. He likewise explains how each claim is anticipated or, at a minimum, rendered obvious by the prior art.

B. Grounds of Invalidity for Challenged Claims 36 and 38 Based on Browne as a Primary Reference

Ground 1: Browne Anticipates Challenged Claims 36 and 38

Browne was not cited to the USPTO and not considered by the examiner during prosecution of the '791 patent. It is prior art under at least 35 U.S.C. § 102(a) and anticipates each of claims 36 and 38 of the '280 patent. [9]

Browne describes the Bulk File Distribution ("BFD") package developed by researchers at the University of Tennessee and Bell Laboratories as part of an effort to make scientific software easily accessible over the Internet. (Browne at 1, 6; Ex. 1002.) The BFD package is based on the concept of a "virtual repository,"

[9]As indicated *supra*, note 1, Petitioners may rely on earlier versions of this article (Ex. 1006 and 1007), alone or in combination with other references cited in this petition, if the Patent Owner alleges an earlier priority date of the challenged claims,

which is a distributed network of physical software repositories, each residing on a different file server. (Browne at 1–2; Ex. 1002.) Browne focuses, in particular, on the mirroring of a file to multiple secondary servers: "copies of popular software packages may be mirrored by a number of sites to increase availability (e.g., if one site is unreachable, the software may be retrieved from a different site) and to prevent bottlenecks." (Browne at 2; Ex. 1002.)

Like the '280 patent, Browne begins by discussing the shortcomings of context- or location-dependent file identifiers. At the time, a virtual repository could be implemented by using a Uniform Resource Locator (URL) to identify each file. (Browne at 2; Ex. 1002.) A URL can be used to specify (i) a transfer protocol, (ii) a location, such as a web server, and (iii) a file name. For example, "http://www.netlib.org/index.html" is a URL that identifies a resource to be accessed (i) using the HyperText Transfer Protocol (HTTP), (ii) at an Internet location "www.netlib.org," and (iii) with file name "index.html." (*See, e.g.*, T. Berners-Lee et al., "Uniform Resource Locators (URL)," Internet RFC 1738 (Dec. 1994) ; Ex. 1018.)

The Browne authors identify several problems with the use of location-based identifiers, such as URLs, to access virtual software repositories. One of their primary concerns was "ensuring the consistency and currency of mirrored copies." (Browne at 2; Ex. 1002.) For example, if the content of a file is updated, the

corresponding URL at the mirror site may become outdated. (Browne at 2; Ex. 1002.) Moreover, a URL can only identify a single location; if a virtual software repository offers multiple mirrored copies of the same file, each copy must be given its own unique URL. (Browne at 2; Ex. 1002.)

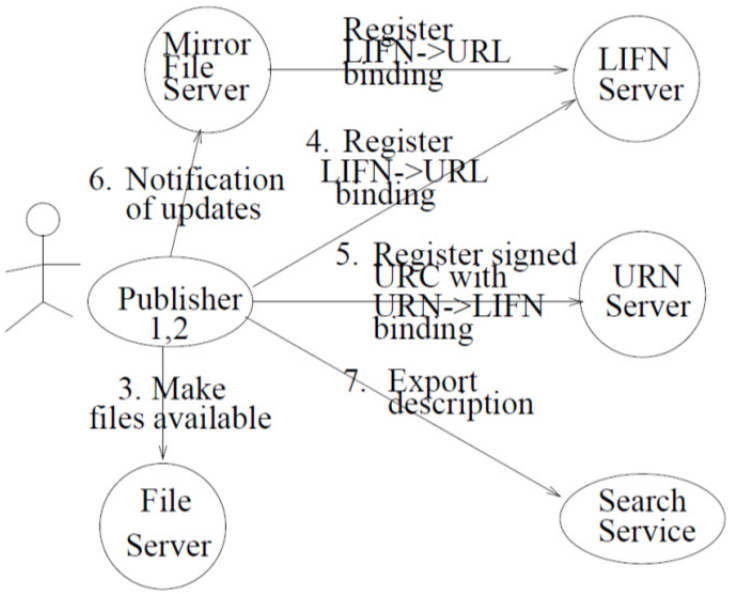
In order to address these shortcomings, Browne adopts the same solution that would be later proposed in the '280 patent: associating a unique identifier with the ***contents*** of a file, rather than with the location of the file. (Browne at 3; Ex. 1002.) Indeed, Browne even refers to its file names as "location independent," the same terminology later used in the '280 patent: "the identity of a data item is ***independent*** of its name, origin, ***location*** . . . ." ('280 patent, col. 3, ll. 32-33 (emphasis added); Ex. 1001.) In the BFD package, the identifier is called a Location Independent File Name, or LIFN. (Browne at 3; Ex. 1002.)

Instead of identifying a physical location, a LIFN uniquely identifies immutable content, i.e., a fixed sequence of bytes. (Browne at 3; Ex. 1002.) As a consequence, two files with identical content will have the same LIFN, even if they are stored on two different servers and are given different names.

In the preferred approach of Browne, the LIFN is computed as the MD5 hash of the contents of the file. (Browne at 6; Ex. 1002.) The general syntax for the LIFN is "lifn:netlib:<signature>", referencing the file access protocol ("lifn," similar to the "http" protocol identifier in a URL), the server handling the request

("netlib"), and the unique MD5 hash used to identify a file ("<signature>").[10] (Browne at 4, 6; Ex. 1002.) The MD5 algorithm provides a substantially unique fingerprint, meaning that two files with identical content will always have the same MD5 fingerprint, even if they are located on different servers, and even if the server administrators give them different names.

Like the '280 patent, Browne uses these hash-based identifiers to store, request, and obtain mirrored copies of files. Figure 2 illustrates the steps involved in "publishing" a resource, such as a file, on Browne's distributed, mirrored file repository:



[10] Dr. Clark confirms that the MD5 <signature> component of the LIFN is the data or file identifier. The first part (protocol) identifies a protocol, and the second part (server) identifies the server used for finding the location of a file. (Clark Decl. ¶ 20; Ex. 1009.)

A "publisher" places a new file on one or more master file servers, and notifies the mirror sites of the existence of the new file; it also provides the search servers (*e.g.*, the LIFN server, discussed below) with the necessary indexing information. (Browne at 4; Ex. 1002.)

Like the '280 patent, Browne uses hash-based identifiers to request and obtain copies of these mirrored files. Specifically, a client computer sends a request to a LIFN server including an MD5 hash (i.e., the LIFN <signature>) of the desired file to be accessed. (Browne at 4–5; Ex. 1002.) In response, the LIFN server returns a list of file servers that store a copy of the file associated with that LIFN.[11] (Browne at 4–5; Ex. 1002.) To be clear, this mechanism is just like the True File Registry (TFR) of the '280 patent, which receives a True Name and provides a list of file servers that store a copy of that file. ('280 patent, col. 34, ll. 25-31; Ex. 1001.) Thus, like the '280 patent, the LIFN server determines where in the system the file associated with the LIFN is stored. The client can then use the LIFN to request and obtain the file from one of the identified file servers.

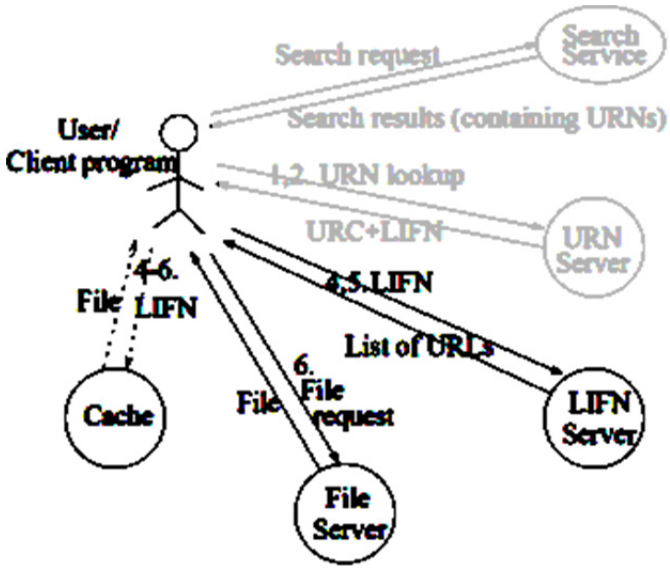
[11] Specifically, a LIFN database associates a LIFN with a list of locations corresponding to that LIFN. (Browne at 6.) Since multiple locations may be used to store the file, mirroring of content is easy and increases reliability of the system. (Browne at 2.)

(Browne at 4–5; Ex. 1002.) Again, the request includes an MD5 hash (*i.e.*, the LIFN <signature>) of the desired file.[12] (Browne at 6; Ex. 1002.)

Figure 3 of Browne (edited for clarity) illustrates the relevant steps in accessing a file, numbered 4 through 6.[13] At step 4, the client accesses a LIFN server, as described above, providing a LIFN for the desired file including an MD5 <signature> uniquely identifying the file of interest; at step 5, and as described above, the LIFN server provides a list of locations (URLs) for servers that store the particular file of interest; finally, at step 6, the identifier is used to access the actual file from one of the locations. (Browne at 5; Ex. 1002.)

[12] In order to accept a request based on a content-based identifier rather than a location-based file name, a file server "create[s] a directory that aliase[s] the ascii form of the MD5 signatures to the actual file locations." (Browne at 6; Ex. 1002.)

[13] Steps 1, 2 and 3 (grayed out in the figure) relate to another aspect of the BFD package, the use of Uniform Resource Names or URNs. The use of URNs is not required in order to use LIFNs. (*See* Browne at 4, 6; Ex. 1002.)



As set forth in detail in the claim chart (Ex. 1034),[14] and as confirmed by Dr. Clark (Clark Declaration at ¶¶ 17-21; Ex. 1009), Browne anticipates each of claims 36 and 38 of the '280 patent. For example, claim 36 recites:

> 36. A method of delivering a data file in a network comprising a plurality of processors, some of the processors being servers and some of the processors being clients, the method comprising:
>
> > storing the data file is[sic] on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server; and

[14] Ex. 1034 is from another proceeding and provides information supporting where Browne, Ex. 1002, discloses the limitations of the challenged claims.

> responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client.

('280 patent, col. 43, ll. 54-64; Ex. 1001.)

As discussed above, Browne discloses a network of processors including client and server processors. (Clark Decl., ¶ 18, 19; Ex. 1009, Browne at 1, 4, Fig. 1; Ex. 1002.) With respect to the limitation of "storing the data file . . . on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server," data files are stored in mirrored file repositories, with each file stored on a primary server and on multiple mirrored servers. (*Id.,* Browne at 2, 4, Fig. 3; Ex. 1002)

With respect to the limitation of "responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client," Browne associates each file with a LIFN, which can be used to obtain the file from a remote server on the network. (Clark Decl., ¶ ¶ 20-21; Ex. 1009, Browne at 3-5; Ex. 1002.) Specifically, a client can send a request to a LIFN server, the request including an MD5 hash of the contents of the file (i.e., the LIFN <signature>). (Clark Declaration at ¶ 21; Ex. 1009, Browne at 4, 6; Ex. 1002.) In response, the LIFN server provides the client with a list of mirror sites that store a copy of the data file corresponding to that LIFN. (*Id.*, Browne at 6; Ex. 1002.) The client can then send a request to one of the mirror

sites, this request also including the MD5 hash (i.e., the LIFN <signature>). (*Id.,* Browne at 4, 5, Fig. 3; Ex. 1002.) In response to the request, the data file corresponding to the LIFN is provided to the client. (*Id.,* Browne at 6; Ex. 1002.)

Statements the patent applicant made during prosecution about claims 36 and 38 demonstrate the invalidity of the claims. When the claims stood rejected by the examiner as not supported by the specification, the patent applicant described specifically what in the specification supported the claims and the information that the applicant relied upon to support the claim is demonstrably in the prior art. In fact, Browne teaches the very same thing as the applicant identified as supporting the claim.[15] For example, in identifying support of the preamble of claim 36 the applicant said that the "'two processors 102 may be in a client/server, client/client, or a server/server relationship. . .' pg. 8 lines 21-24" and also cited Fig. 1 and pg. 8, lines 8 to pg. 9, line 3. (Response, Aug. 22, 2001 at 38; Ex. 1019.) Browne discloses the same type of distribution of data files over a network that comprises a plurality of LIFN servers, mirror and cache sites (servers), and clients, all of which are processors. The processors may be in a client/server relationship (e.g., a client and a LIFN server, or a file server) or in a server/server relationship (e.g., a mirror site and a LIFN server). (Browne at 4, Fig. 2; Ex. 1002.)

---

[15] The disclosure of Browne is only representative. A more complete discussion is provided below and in the attached claim chart (Ex. 1034).

Statements about what supports the body of claim 36 are similarly revealing. The applicant said that the "'storing the data file" limitation was disclosed in the "Mirror True File" which is '. . . use[d] to ensure that files are available in alternate locations.' Pg. 50, lines 16-18." (Response, Aug. 22, 2001 at 38; Ex. 1019.) Browne discloses a data file stored on a file server in the network and replicated onto a plurality of other servers on the network, such as mirror sites and/or cache sites. Browne discloses the same thing- the replicated servers are used to ensure that files are available in alternate locations. (Browne at 2, 4–5; Ex. 1002.) Finally, for the "responsive to a client request" limitation, the applicant relied upon statements about "Realize True File from Location, pg. 29, lines 12 to pg. 30, line 5" and "Request True File, pg. 46, lines 7-24." and "'functions include MD4, MD5 and SHA' pg. 23, lines 11-12. These are known as hash functions." (Response, Aug. 22, 2001 at 38; Ex. 1019.) Brown discloses the very same thing - a client request for a data file, including a hash of the contents of the file. The hash function is an MD5 hash. (Browne at 5–6; Ex. 1002.) Having cited the specification to the Patent Office to show support for the claims, the Patent Owner cannot now run away from these statements when the same statements are clearly found in the prior art. Consequently, to the extent the specification of the '280

patent supports the challenged claims, Browne anticipates them for the same reasons.[16]

A similar analysis applies to the other challenged claim of the '280 patent, and the analysis for claim 36 is incorporated by reference. Claim 38 has the same limitations as claim 36, except that claim 38 refers to a client request for a data file "including a value determined as a given function of the contents of the data file," rather than "a hash of the contents of the data file." Browne discloses these requirements, because the MD5 hash included in the LIFN <signature> is a value determined as a function of the contents of the data file. (Clark Del., ¶¶ 19-21; Ex. 1009, Browne at 5, 6; Ex. 1002.)

Ground 2: Challenged Claims 36 and 38 are Unpatentable as Obvious in view of Browne in combination with Langer

In the event PersonalWeb contends that Browne does not satisfy the claim limitations of "the request including a hash of the contents of the data file" (claim 36) or including "a value determined as a given function of the contents of the data file" (claim 38), a person of ordinary skill in the art would have found it obvious to combine Browne with another reference such as Langer, in order to use a hash of

[16] The Patent Owner is estopped from taking a different position on the claim charts. The above-referenced representations made to the USPTO during prosecution of the '280 patent secured allowance of the claims, and the doctrine of estoppel thus precludes the Patent Owner from taking an inconsistent position.

the contents of the data file (*e.g.*, a value determined as a given function of the contents of the data item) as a unique identifier.[17] Dr. Clark confirms that such a combination would have been desirable to provide a substantially unique identifier, with negligible chances of collision. (Clark Decl., ¶22; Ex. 1009.) In addition, as disclosed in Langer, a hash can be generated locally from the contents of the file itself, whereas other types of identifiers (e.g., a serial number) have to be assigned by a centralized server, in order to prevent duplications. (*Id.,* Langer at 4; Ex. 1003.) The modified BFD package would meet the limitation of "the request including a hash [or given function] of the contents of the data file." (*Id.*) The application of Langer's hash identifier to Browne would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results, and therefore it would have been obvious to a person of ordinary skill in the art. (*Id.*) This analysis is further supported by the claim chart (Ex. 1035).

C. Grounds of Invalidity for Challenged Claims 36 and 38 based on Woodhill as a Primary Reference

Ground 3: Woodhill Anticipates Challenged Claims 36 and 38

[17] Alternatively, the same result would be obtained by combining Woodhill.

Woodhill was not cited to the USPTO and not considered by the examiner during prosecution of the '280 patent. It is prior art under at least 35 U.S.C. § 102(e) and anticipates each of claims 36 and 38 of the '280 patent.

Woodhill discloses a distributed storage management system, with mechanisms for storing multiple backup copies of files and later restoring (*i.e.*, accessing) those files. (Woodhill at col. 2, ll. 39-49; Ex. 1005.) The system includes a network of processors, including client and server processors. Figure 1 of Woodhill depicts this network:

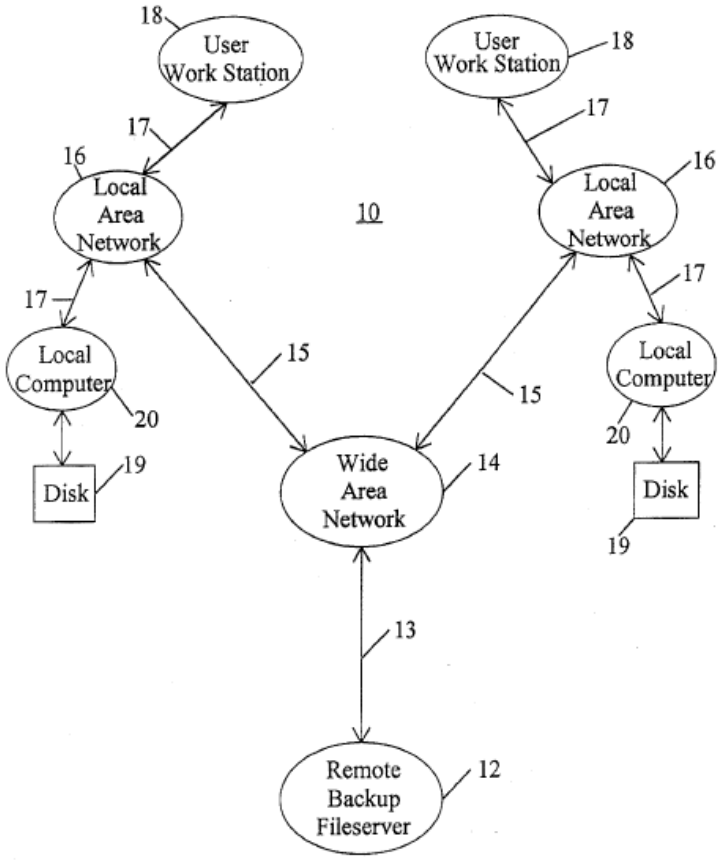


FIG. 1

During a backup cycle, Woodhill backs up the contents of each disk 19 of each local computer 20 in two places: (1) on another local computer 20 accessible through a local area network 16, and (2) on the remote backup file server 12. (*Id*. at col. 9, ll. 30-38; Ex. 1005.) Thus, once a backup cycle has completed, the network stores three copies of each file: the original copy on the local computer 20, a backup copy on a second local computer 20, and a backup copy on the remote backup file server 12. A local computer 20 can access the backed-up files, either from a different local computer 20, or from the remote backup file server 12, as needed. (*Id*. at col. 9, ll. 39-44; Ex. 1005.)

As Woodhill explains, binary objects available on the "local area network 16 can be restored very quickly while the much greater storage capacity on the remote backup file server 12 ensures that at least one copy of every binary object is stored and that a disaster that destroys an entire site would not destroy all copies of that site's data." (*Id*. at col. 9, ll. 40-44; Ex. 1005.) In this way, local computers may act as servers to each other for rapid access to files under normal operating conditions, while a remote backup file server ensures that data will remain available even if those local computers are destroyed.

Before backing up files at these locations, Woodhill breaks the files into "binary objects" having a size of one megabyte or less.[18] (*Id*. at col. 4, ll. 12-30; Ex. 1005.) To identify binary objects, a Binary Object Identification Record, including a Binary Object Identifier, is determined for each binary object. (*Id*. at col. 7, l. 60 – col. 8, l. 1; Ex. 1005.) These Binary Object Identifiers are calculated from "the contents of the data" of the corresponding binary object, and include a hash of the contents of the object. (*Id*. at col. 8, ll. 40-41; Ex. 1005.) As Woodhill makes clear, the "***critical feature to be recognized in creating a Binary Object Identifier 74 is that the identifier should be based on the contents of the binary object*** so that the Binary Object Identifier 74 changes when the contents of the binary object changes." (*Id*. at col. 8, ll. 58-62, emphasis added; Ex. 1005.)

---

[18] Woodhill "views a file as a collection of data streams." (Woodhill at col. 4, ll. 14-15; Ex. 1005.) Each data stream is "a distinct collection of data within the file that may be changed independently from other distinct collections of data within the file." (*Id*. at col. 4, ll. 15-18; Ex. 1005.) Woodhill "divides each data stream into one or more binary objects. If the size of the data stream is equal to or less than a previously defined convenient maximum binary object size (current one (1) megabyte), then a single binary object represents the data stream." (*Id*. at col. 4, ll. 22-26; Ex. 1005.) Accordingly, as Dr. Clark confirms, a file with only one corresponding data stream may also have only one binary object. (Clark Decl., ¶ 25 fn. 5; Ex. 1009.)

Accordingly, "the possibility of two different binary objects being assigned the same Binary Object Identifier 74 [is] very small," and each Binary Object Identifier 74 " uniquely identif[ies] a particular binary object." (*Id*. at col. 8, ll. 33-36; Ex. 1005.)

Figure 3 of Woodhill depicts the Binary Object Identifier portion of the Binary Object Identification Record. The Binary Object Identifier is 128 bits long and includes a Binary Object Hash. (*Id*. at col. 7, l. 64 – col. 8, l. 4; Ex. 1005.) This Binary Object Hash is "calculated against the contents of the binary object taken one (1) word (16 bits) at a time":

```
HASH = (initialized value)
for each word (16 bits) of the binary object:
   rotate current HASH value by 5 bits
   HASH = HASH + 1
   HASH = HASH + (current word (16 bits) of binary object)
end loop
```

(*Id*. at col. 8, ll. 21-31; *see also* the hash algorithm reproduced below; Ex. 1005.) Thus, Binary Object Identifiers 74, like the True Names of the '280 patent, are based on the contents of the binary objects and include a hash of the contents of the objects.

Once the binary objects forming a file have been backed up, Woodhill "perform[s] self-audits on a periodic basis to ensure that the binary objects that have been backed up can be restored." (*Id*. at col. 18, ll. 12-13; Ex. 1005.) This

process proceeds by a client "initiat[ing] a restore of a randomly selected binary object identified by a Binary Object Identification Record" including the Binary Object Identifier with a Binary Object Hash value. (*Id*. at col. 8, ll. 1-32, col. 18, ll. 17-19; Ex. 1005.) In response to this request, "the selected binary object is restored from either a compressed storage file 32 residing on one of the disk drives 19 of one of the local computers 20 or from the remote backup file server 12." (*Id*. at col. 18, ll. 20-23; Ex. 1005.) Woodhill accordingly provides copies of requested binary objects in response to a request for the binary object, the request including a hash of the contents of the binary object.

As set forth in detail in the claim chart (Exhibit 1032), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 23-27; Ex. 1009), Woodhill anticipates each of claims 36 and 38 of the '791 patent. For example, for claim 36, quoted above, Woodhill discloses a network of processors including client and server processors. (Clark Decl., ¶¶ 24, 25; Ex. 1009, Woodhill at col. 3, ll. 6-31, Fig. 1; Ex. 1005.) With regard to "storing the data file is [sic] on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server," Woodhill stores binary objects (i.e., data files) on a local computer (i.e., the first server) and stores copies of the objects on at least one other local computer and on the remote backup file server (i.e., the set of servers in the network distinct from the first server). (*Id.,* Woodhill at col. 9, ll. 42-44, col. 4, ll. 14-26, Fig. 1; Ex.

1005.) In fact, the "Mirror True File" mechanism – which the applicants cited as support for this limitation during prosecution of the '280 patent – operates just like Woodhill. (*See* Response at p. 38; Ex. 1019.) The '280 patent states that the "Mirror True File" mechanism "is used to ensure that files are available in alternate locations in mirror groups or archived on archival servers." ('280 patent, col. 26, ll. 20-23; Ex 1001). As Dr. Clark confirms, Woodhill's backup file server operates as an archival server of precisely the same nature. (Clark Decl., ¶ 25; Ex. 1009; Woodhill at col. 9, ll. 42-44; Ex. 1005.)

With regard to "responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client," Woodhill discloses that a local computer (i.e., a client) can request that a binary object be restored. (Woodhill at col. 10, ll. 27-32; Ex. 1005.) As Dr. Clark confirms, such a request includes a Binary Object Identifier with a hash of the contents of the requested binary object. (Clark Decl., ¶ 26; Ex. 1009.) In response to such a request, the binary object is provided to the local computer. (*Id.*)

A similar analysis applies to the other challenged claim of the '280 patent, and the analysis for claim 36 is incorporated by reference. Claim 38 refers to a client request for a data file "including a value determined as a given function of the contents of the data file," rather than "a hash of the contents of the data file."

Woodhill discloses either of these requirements, because the hash included in the Binary Object Identifier is a value determined as a function of the contents of the data file. (Clark Decl., ¶¶ 25-26; Ex. 1009; Woodhill at col. 7, l. 60- col. 8, l. 65, Figs. 3, 5A and 5J; Ex. 1005.)

Although Woodhill was not considered by the examiner during prosecution of the '280 patent, the applicants conceded the relevant features of Woodhill in prosecuting later patents in the same patent family. For example, the applicants admitted, while prosecuting related U.S. Patent No. 7,945,539, that "Woodhill describes a system for backing up and restoring data in a network" in which "[l]ocal computer files are backed up to a remote backup server via a wide area network (WAN). ***One*** copy of each file is made on the backup server, and ***one other copy*** of the file is made on another 'local' computer (i.e., another computer on the same local area network (LAN))." (Response to Non-Final Office Action dated October 5, 2009 in prosecution of Application Serial No. 11/980,679, at p. 16) (emphasis added); Ex. 1037). These statements were made in an effort to incorrectly argue that Woodhill does not describe computers in a peer-to-peer (P2P) relationship (a limitation absent from the '280 challenged claims). (*Id*.)

The Patent Office similarly found during prosecution of another family member of the '280 patent that "Woodhill teaches . . . initiat[ing] a restore request for a selected Binary Object identified by a Binary Object Identification Record to

a remote backup file server," and that "Woodhill teaches the Binary Object Identifier is calculated using a Hash Function." (*Id*.)

Ground 4: Challenged Claims 36 and 38 are Unpatentable as Obvious in view of Woodhill

In the event that PersonalWeb contends that Woodhill does not meet the claim limitation of "storing the data file is [sic] on a first server and storing copies of the data file on a set of servers distinct from the first server," a person of ordinary skill in the art would have found it obvious to modify Woodhill to meet that limitation. (Clark Decl., ¶¶ 28-29; Ex. 1009.) First, distributing files in a network including many servers was well known in the art. For example, in response to a rejection of all pending claims of the '280 patent during prosecution, the applicants merely amended the claims "to clarify that the identifier determined for a data identifier is context sensitive, i.e., is based on the content of the data or files." (Response to Non-Final Office Action, August 22, 2001 in prosecution now issued as the '280 patent, at p. 24; Ex. 1019.) The applicants did not dispute that the cited prior art taught any claimed file distribution elements. (*See id*. at pp. 24-26). Second, it would have been obvious to add an additional remote backup file server or servers to Woodhill's system for additional data security (e.g., in the event that the single remote backup file server was destroyed concurrently with a local computer on which a binary object is backed up). (Clark Decl., ¶ 29; Ex. 1009.) Adding additional remote backup file servers to Woodhill would constitute

applying a known technique (i.e., adding extra redundancy) to a known device, method, or product ready for improvement to yield predictable results, and therefore it would be obvious to a person of ordinary skill in the art exercising ordinary creativity. (*Id.*) This analysis is further supported by the claim chart (Ex. 1032).

D. Grounds of Invalidity for Challenged Claims 36 and 38 based on the ESM Manual as a Primary Reference

Ground 5: The ESM Manual Anticipate Challenged Claims 36 and 38

The ESM Manual was not cited to the USPTO and not considered by the examiner during prosecution of the '280 patent. It is prior art under at least 35 U.S.C. § 102(b) and anticipates each of claims 36 and 38 of the '280 patent.

The ESM Manual describes the Enterprise Storage Manager (ESM) product. ESM provides a corporate network backup solution that "protects data by completely automating the backup process, by saving the data in two locations, on the LAN and at the corporate data center, and by automatically auditing the data as it is saved." (ESM Manual at 2-1; Ex. 1026.) Figure 2-1 of the ESM Manual illustrates this "dual" backup process:

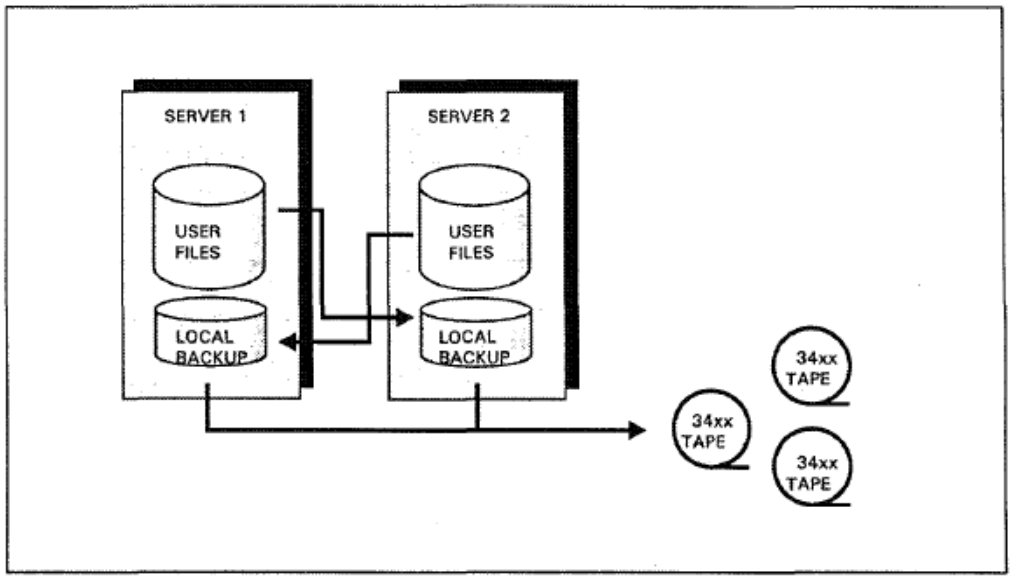


Figure 2-1. Dual Backup Process.

During this process, ESM backs up user files in two locations, in addition to the local computer: (1) on LAN disk storage on a local server and (2) on a mainframe server at the corporate data center that uses backup tapes. (ESM Manual at 2-1 – 2-2, 4-4; Ex. 1026.) The manual comments that storing user files at the corporate data center is advantageous because, by using "high-speed, high-reliability automated tape systems, mainframes can locate and retrieve data files from tape much faster than LANs, and are physically secure in ways that are difficult, if not impossible, to duplicate in typical office environments." (ESM Manual at 2-2; Ex. 1026.) Thus, using the example of Figure 2-1, this process leads to three total copies of the "User Files": the original copies stored on Server 1, the "Local Backup" stored on Server 2, and the tape backups stored by the corporate data center.

To perform these backup operations, the ESM manual discloses "decompos[ing] files into binary objects, or BOBs."[19] (ESM Manual at 3-3 – 3-4; Ex. 1026.) The manual further discloses a "name of a BOB, the BOBID" that "is calculated from the contents of the BOB itself." (ESM Manual at 3-4; Ex. 1026.) More specifically, each BOBID includes a 32-bit hash value representing a hash of the contents of the BOB. (ESM Manual at 3-4; Ex. 1026.)

To request that a file be restored, a client computer submits a restore request using the BOBID: "ESM's architecture enables all restore requests to arrive as BOBID requests." (ESM Manual at 3-9; Ex. 1026.) This provides an element of security because, "[t]o retrieve any data file, the user must already know the identity of the data, an identity only available to the ESM backup process." (ESM Manual at 3-9; Ex. 1026.)

As set forth in detail in the claim chart (Exhibit 1033), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 30-38; Ex. 1009), the ESM manual anticipates each of claims 36 and 38 of the '791 patent. For example, for claim 36, the ESM manual discloses a network of processors including client and server processors. (ESM Manual at Figure 4-1; Ex. 1026.) With regard to "storing the data file is [sic] on a

---

[19] For large files, the data portion is broken up into a number of BOBs of 1 megabyte. As Dr. Clark confirms, the data of small files (e.g., less than 1 megabyte), only a single BOB will exist. (Clark Decl. at ¶ 36; Ex. 1009.)

first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server," the ESM manual discloses that the BOBs originate on one server (i.e., the first server) and that copies are stored on both a second server and at the corporate data center (i.e., the set of servers in the network distinct from the first server). (Clark Decl., ¶¶ 32-35; Ex. 1009, EMS Manual at 2-1, Fig. 2-1; Ex. 1026.)

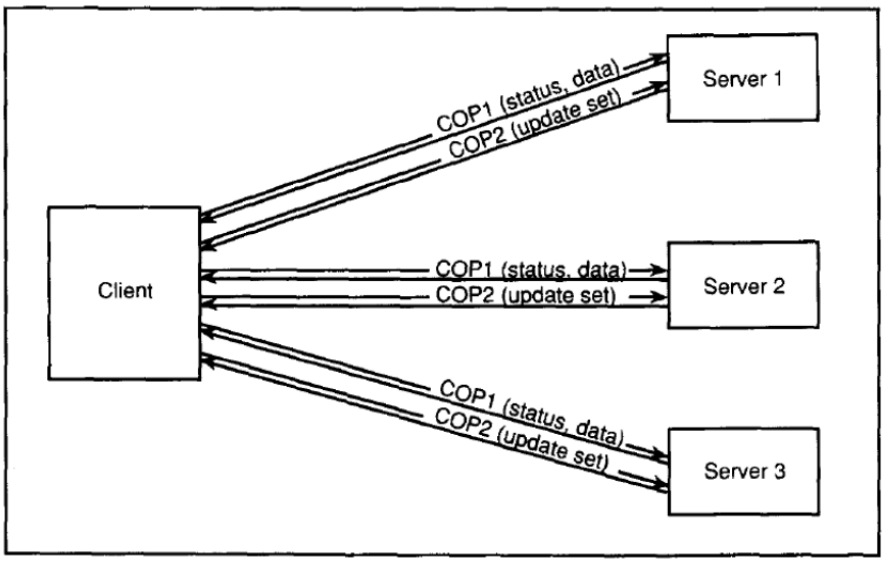
With regard to "responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client" (claim 36) and "responsive to a client request for the data file, the request including a value determined as a given function of the contents of the data file, providing the data file to the client" (claim 38), a client computer may request that a BOB be restored in a request including a BOBID. (Clark Decl., ¶¶ 36-37; Ex. 1009, EMS Manual at 3-3, 3-4; Ex. 1026.) Dr. Clark confirms that the BOBIDs include a hash of the contents of the requested BOBs. (Clark Decl., ¶ 36; Ex. 1009.) In response, the requested BOBs are provided to the client computer. (Clark Decl., ¶ 37; Ex. 1009, EMS Manual at 3-9, 5-3; Ex. 1026.)

E. Grounds of Invalidity for Challenged Claims 36 and 38 based on Satyanarayanan as a Primary Reference

Ground 6: Satyanarayanan in combination with Langer Renders Challenged Claims 36 and 38 Obvious

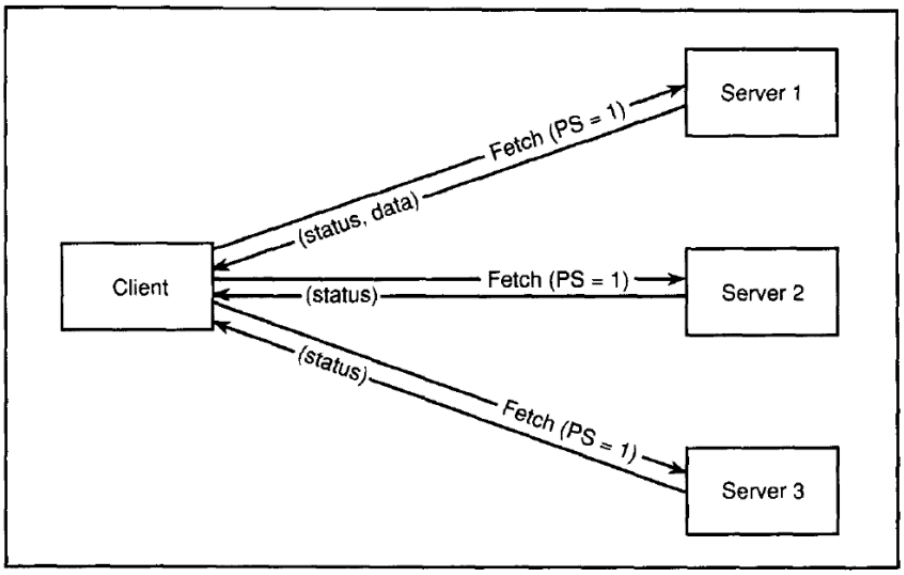
Satyanarayanan was not cited to the USPTO and not considered by the examiner during prosecution of the '791 patent. It is prior art under at least 35 U.S.C. § 102(b) and renders obvious each of claims 36 and 38 of the '280 patent.

Like the '280 patent, Satyanarayanan teaches storing multiple copies of data files on a set of servers in a network, and providing client workstations with access to those data files. Specifically, Satyanarayanan's *Coda* file system maintains multiple replicas of data files on a group of servers (Satyanarayanan at 16; Ex. 1029.) Satyanarayanan discusses the example of a group of three servers, one of which functions as the preferred server. (Satyanarayanan at 16; Ex. 1029.) When writing a file, a client writes ***one*** copy to the preferred server and ***two*** more copies to the other servers, as depicted in Figure 6 of Satyanarayanan, reproduced below:



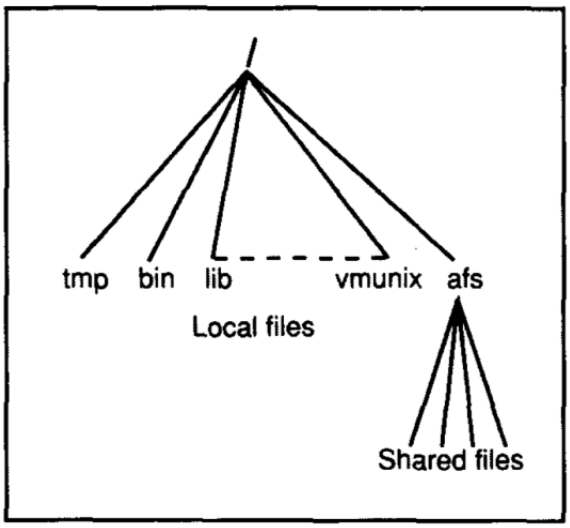
In order to read a data file, a client sends a "fetch" request to the preferred server. (Satyanarayanan at 16; Ex. 1029.) In response, the preferred server

provides the data file to the client, as depicted in Figure 5 of Satyanarayanan, reproduced below:



The other two servers provide the client with status information. (Satyanarayanan at 16; Ex. 1029.)

The Coda system uses server replication to increase the availability of shared data, but from a user's viewpoint it implements the standard functionality of a file system. (Satyanarayanan at 10, 15; Ex. 1029.) For example, a client's view of the file system has the conventional Unix "tree" structure, as depicted in Figure 2 of Satyanarayanan, reproduced below:



As the diagram indicates, files are accessible through a conventional pathname, *e.g.*, shared files are identified by a pathname starting with "/afs." (Satyanarayanan at 10; *see also id.* at 10–12, 18 (discussing pathnames); Ex. 1029.)

Although Satyanarayanan did not use hash-based identifiers to request and obtain files, multiple prior art references, including Langer and Kantor in addition to Browne, Woodhill, and the ESM manual, each confirm that identifying files with hashes of the file contents also was known well before the '280 patent. For example, Langer discloses "a unique identifier for each file which is independent of location," implemented as "a hash function on the entire contents of the file." (Langer at 3–4; Ex. 1003.) Moreover, Langer expressly recognizes that a file identifier based on a hash could be substituted for a file identifier based on a

pathname (such as that used in Satyanarayanan), for example, to identify a file to be downloaded from an FTP site. (Langer at 4–5; Ex. 1003.)

As set forth in detail in the claim chart (Exhibit 1036), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 39-44; Ex. 1009), Satyanarayanan, in combination with Langer, renders each of claims 36 and 38 obvious. For example, Satyanarayanan discloses a network of processors including client and server processors as recited in both claims 36 and 38. (Satyanarayanan at 16, Fig. 6; Ex. 1029.) With regard to "storing the data file is [sic] on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server," recited in claims 36 and 38 Satyanarayanan discloses a replicated file system whereby a preferred server stores a data file, and two backup servers in the network distinct from the primary server store copies of the data file. (Clark Decl., ¶41; Ex. 1009, Satyanarayanan at 16, Fig. 6; Ex. 1029.)

With regard to "responsive to a client request for the data file, the request including a hash of the contents of the data file, causing the data file to be provided to the client," recited in claims 36 and 38 Satyanarayanan discloses providing a file to a client in response to the client's "fetch" request for the file. (Clark Decl., ¶ 42; Ex. 1009, Satyanarayanan at 10-12, Fig. 5; Ex. 1029.) Although Satyanarayanan did not use file identifiers including a hash of the contents of the data file, Langer used hash-based identifiers for precisely the same reason as that of the '280 patent,

i.e., content-based, location-independent substantially unique identification of a file. As Dr. Clark confirms, a person of ordinary skill would have found it obvious to apply the teachings of Langer to Satyanarayanan to enable client requests for files that use content-based identifiers including a hash of the contents of a file. (Clark Decl., ¶ 43; Ex. 1009.) In fact, Langer provides express motivation to make this combination, by disclosing that the use of a unique, location-independent identifier allows a client to automatically access a file from the nearest location. (*Id.*; *see also* Langer at 3–4; Ex. 1003.) Such a combination would have been an obvious application of Langer's known technique to the known device of Satyanarayanan, ready for improvement, to yield the predictable result of improving file access using unique identifiers. (*See* Langer at 3–4; Ex. 1003.) As such, the combination would have been the mere combination of prior art elements, according to known methods, to yield predictable results. (Clark Decl., ¶ 43; Ex. 1009.) Further, design incentives and market forces would have prompted the application of Langer to Satyanarayanan, because the resulting system would have achieved the desired uniqueness of the identifiers at a minimum of complexity. (*Id.*) Finally, the combination would have required no more than ordinary creativity, taking into account the inferences and creative steps that a person of ordinary skill in the art would employ. (*Id.*)

Ground 7: Satyanarayanan in combination with Kantor Renders Challenged Claims 36 and 38 Obvious

To the extent Satyanarayanan does not explicitly disclose "the request including a hash of the contents of the data file," it also would have been obvious to combine Satyanarayanan with Kantor. Like the '280 patent, Kantor discloses content-based identifiers based on "a hash of the contents of the data file," called "contents-signatures." (*See* Kantor at 6; Ex. 1004.) As Dr. Clark confirms, these contents-signatures include a hash of the contents of the data files. (Clark Decl., ¶ 44; Ex. 1009.) Kantor expressly teaches the benefits of using these content-based identifiers, which "[do] not depend on file names, dates, order of collection, nor method nor amount of compression." (Kantor at 2-3; Ex. 1004.)

As set forth in detail in the claim chart (Exhibit 1036), and as confirmed by Dr. Clark (Clark Decl., ¶ 39-41, 44; Ex. 1009.), Satyanarayanan, in combination with Kantor, renders each of claims 36 and 38 obvious. For example, as described above for Ground 5 and incorporated by reference herein, Satyanarayanan discloses a network of processors including client and server processors and "storing the data file is [sic] on a first server in the network and storing copies of the data file on a set of servers in the network distinct from the first server," and "responsive to a client request for the data file," the request including a hash of the contents of the data file, causing the data file to be provided to the client." Although Satyanarayanan did not use file identifiers including a hash of the contents of the data file, Kantor used hash-based identifiers for precisely the same

reason as that of the '280 patent, i.e., content-based, location-independent substantially unique identification of a file. (Kantor at 6-11; Ex. 1004.) As Dr. Clark confirms, a person of ordinary skill would have found it obvious to apply the teachings of Kantor to Satyanarayanan. (Clark Decl., ¶ 45; Ex. 1009.) Kantor provides an express motivation to do so, by disclosing that the use of a substantially unique content-based identifier allows a client to lookup a file at a remote file server, i.e., an electronic Bulletin Board System (BBS). (Kantor at 97; Ex. 1004.) Even the '280 patent teaches posting True Name files to computer bulletin boards. (*See* '280 patent, col. 35 ll. 56-59; Ex. 1001.) Dr. Clark confirms that such a combination would have been desirable to provide an additional means for clients to request data files from the Coda replicated file system. (Clark Decl., ¶ 44; Ex. 1009.) The modified Coda system would meet the limitation of "the request including a hash of the contents of the data file" (claim 36) and "the request including a value determined as a given function of the contents of the data file" (claim 38). (*Id.*) The combination of Satyanarayanan with Kantor would have applied Kantor's known technique to the known device of Satyanarayanan, ready for improvement, to yield the predictable result of improving file access using identifiers based on a hash of the contents of the file. (*Id.*) As such, the combination would have been the mere combination of prior art elements, according to known methods, to yield predictable results. (*Id.*) Further, design

incentives and market forces would have prompted the application of Kantor to Satyanarayanan, because the resulting system would have achieved the desired uniqueness of the identifiers at a minimum of complexity. (*Id.*) Finally, the combination would have required no more than ordinary creativity, taking into account the inferences and creative steps that a person of ordinary skill in the art would employ. (*Id.*)

VI. CONCLUSION

Based on the foregoing, it is clear that claims 36 and 38 of the '280 Patent recite subject matter that is anticipated or, at a minimum, obvious. The art cited above was never considered by the original Patent Examiner, and if it had been the '280 patent would not have issued. The Petitioner requests institution of an *inter partes* review to cancel those claims.

Respectfully Submitted,

/David L. Cavanaugh/

David L. Cavanaugh
Registration No. 36,476

CERTIFICATE OF SERVICE

I hereby certify that, on December 15, 2012, I caused a true and correct copy of the foregoing materials:

- Petition for *Inter Partes Review* of U.S. Patent No. 6,415,280
- Exhibits 1001-1037
- Fee Summary Page
- EMC Corp. Power of Attorney
- VMware, Inc. Power of Attorney

to be served via Express Mail on the following attorney of record as listed on PAIR:

Pillsbury Winthrop Shaw Pittman, LLP

PO Box 10500

McLean, Virginia 22102

/David L. Cavanaugh/

David L. Cavanaugh

Registration No. 36,476

Table of Exhibits for U. S. Patent 6,415,280 Petition for Inter Partes Review

Exhibit	Description
1001	U.S. Patent No. 6,415,280
1002	S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," University of Tennessee Technical Report CS-95-278 (Feb. 1995)
1003	Albert Langer, "Re: dl/describe (File descriptions)," post to the "alt.sources" newsgroup on August 7, 1991
1004	F. Kantor, "The Frederick W. Kantor Contents-Signature System Version 1.22," FWKCS122.REF (August 10, 1993)
1005	Woodhill et al., U.S. Patent No. 5,649,196, entitled "System and Method For Distributed Storage Management on Networked Computer Systems Using Binary Object Identifiers,"
1006	S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," http://www.netlib.org/utk/papers/lifn/main.html (Nov. 11, 1994)
1007	K. Moore et al., "An Architecture for Bulk File Distribution," Network Working Group Internet Draft (July 27, 1994)
1008	Chart of Patent Family Members
1009	Declaration of Dr. Douglas W. Clark, PH.D
1010	Banisar et al., The Third CPSR Cryptography and Privacy Conference at 509 (1993)
1011	G. D. Knott, Hashing functions, The Computer Journal (1975), vol. 3, no. 3, p. 265.
1012	R. Rivest, "The MD5 Message-Digest Algorithm,"

Exhibit	Description
	Internet RFC 1321 (Apr. 1992)
1013	McGraw-Hill Dictionary of Scientific and Technical Terms, (4th ed., 1989)
1014	B. Kaliski, "A Survey of Encryption Standards, " IEEE Micro (Dec. 1993)
1015	Rabin, Fingerprinting by Random Polynomials, Center for Research in Computing Technology, Harvard University, Report TR-15-81
1016	U. Manber, "Finding Similar Files in a Large File System", University of Arizona Technical Report (1994)
1017	D.R. McGregor and J.A. Mariani 'Fingerprinting' – A Technique for File Identification and Maintenance, 12 Software Practice & Experience 1165 (1982)
1018	T. Berners-Lee et al., "Uniform Resource Locators (URL)," Internet RFC 1738 (Dec. 1994)
1019	U. S. Patent 6,415,280 Prosecution History, Response (August 22, 2001)
1020	EP Pub. No. EP0826181A1 Prosecution History, Annex to the communication dated May 8, 2009
1021	EP Pub. No. EP0826181A1 Prosecution History, Reply to communication from the Examining Division dated November 18, 2009
1022	EP Pub. No. EP0826181A1 Prosecution History, Annex to the communication dated March 14, 2012
1023	EP Pub. No. EP0826181A1 Prosecution History, Closing of Application dated June 14, 2012
1024	U.S. Patent No. 6,425,280 Preliminary Amendment (Jan. 11, 2001)

Exhibit	Description
1025	U.S. Patent No. 6,425,280 Office Action (June 5, 2001)
1026	Legent Software, Inc., "ESM: Product Information," Legent Corporation (April 1994)
1027	R. Williams, "An algorithm for matching text", posted to the "comp.compression" newsgroup (January 27, 1992)
1028	R. Williams, "An Introduction to Digest Algorithms," Rocksoft (Nov. 1994)
1029	Satyanarayanan, "Scalable, Secure, and Highly Available Distributed File Access," IEEE Computer, vol. 23, no. 5 (May 1990)
1030	P. Alsberg and J. Day, "A Principle for Resilient Sharing of Distributed Resources", Proc. of the 2d International Conference on Software Engineering (1976)
1031	J. Bartlett, "A 'NonStop' Operating System", Proc. of the Eleventh Hawaii International Conference on System Sciences (1978)
1032	Invalidity Claim Chart in view of Woodhill
1033	Invalidity Claim Chart in view of ESM
1034	Invalidity Claim Chart in view of LIFN ("Browne")
1035	Invalidity Claim Chart in view of Langer
1036	Invalidity Claim Chart in view of Satanarayanan
1037	U. S. Patent 7,945,539 Prosecution History, Response to Non-Final Office Action (Oct. 5, 2009)